Perspectives on M2M protocols

A comparative study between different M2M protocols

Mohamed H. Elgazzar
Aljizah, Egypt
mohamedhelgazzar@gmail.com

***Abstract*— In the last years, the Machine-to-Machine (M2M) communications evolved as one of the major trends shaping the development of services in the future Internet. The use of M2M technologies is rapidly increasing in different fields such as Health Care, Automotive, Energy, Consumer Electronics, manufacturing, security and Banking. With that rapid increase different protocols had been developed for M2M communications and device management such as CoAP, MQTT and LwM2M.**

The objective of the paper is to provide synopsis of the different M2M protocols. The paper discusses the pros and cons of each of the protocols and identify their open problems. Up to our knowledge, it is the first study to provide such comparison between the different M2M protocol. The comparison is based on supported functions, network overhead, network reliability and security beside highlighting the different protocol architectures. The paper works as a start point for producing efficient M2M protocols

Keywords—M2M; Machine to Machine; CoAP; OMA DM; MQTT, LwM2M, XMPP, CWMP

I. INTRODUCTION

Cellular network-based machine-to-machine (M2M) communication is fast becoming a market-changing force for wide spectrum of businesses and applications such as smart metering, vending machines, and security. M2M is based on smart devices that function without direction human intervention. Compared to traditional automation technologies, one major difference for this new generation of smart devices is how tightly they are coupled into larger scale service infrastructures. [1] For example, in logistic operations, the vehicle can be tracked with Automatic Vehicle Location (AVL) and uploaded into back-end automatic dispatching and planning system for real-time global fleet management [2].

As a result of the wide-spread and rapid evolution of the smart devices and back-end applications different protocols had been developed in order to serve the M2M communication system. Those protocols includes 1) CoAP which is designed by IETF to enable the manipulation of resources for constrained devices that are capable of connecting to the Internet [3]. 2) Smart M2M which is developed by ETSI and provides specifications for M2M services and applications, and particularly focuses on aspects of Internet of things (IoT) and Smart Cities 3) MQTT (Message Queuing Telemetry Transport) is designed by OASIS and is a publisher/subscriber messaging protocol developed for constrained devices [4]. 4) LwM2M (Lightweight M2M) focuses on service enablement of particular resource constrained devices [5]. The remaining of the paper is organized as follows: Section II presents the M2M communication protocols which are used in the communication between the different M2M components. Section III presents M2M device management protocols which are the protocols used for managing the different remote M2M devices. Section IV presents the security features in each of the presented protocols. Section V the conclusion.

Figure 1 shows the general architecture of M2M applications:

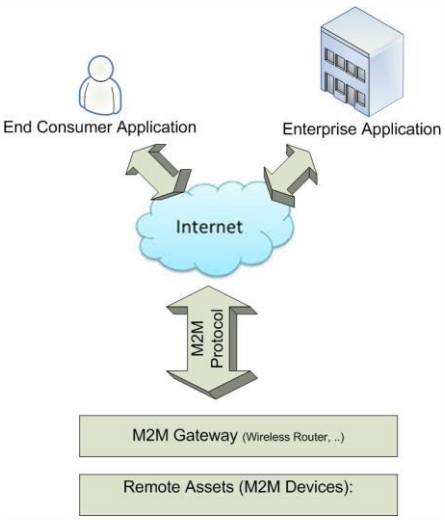

Fig. 1. M2M Architecture

II. M2M COMMUNICATION PROTOCOLS

A. CoAP

CoAP was developed as an Internet standard document RFC 7252 [6]. The interaction model of CoAP is similar to the client/server model of HTTP. However, machine-to-machine interactions typically results in a CoAP implementation acting in both client and server roles. A CoAP request is equivalent to that of HTTP and is sent by client to request an action (using a method code). Unlike HTTP, CoAP deals with these interchanges asynchronously over a datagram-oriented transport such as UDP. This is done logically using a layer of messages that supports optional reliability. CoAP defines four types of messages: 1) Confirmable 2) Non-Confirmable 3) Acknowledgement 4) Reset. Method codes and response codes included in some of these messages make them carry requests

and responses codes. Requests can be carried in confirmable or non-confirmable messages, and responses can be carried on them as well as piggyback in Acknowledgement messages [7]. CoAP overhead is only 4 bytes [8]

Figure 2 shows a comparison between the CoAP and HTTP stacks:

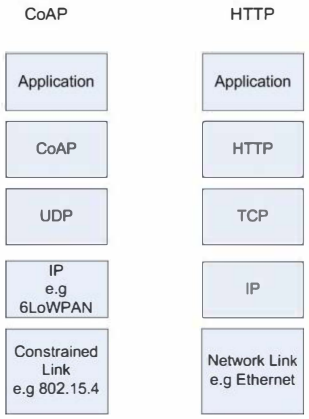

Fig. 2. CoAP vs HTTP protocol stack

Although CoAP provides an architecture for the M2M, a main problem of CoAP is providing only basic congestion control mechanism CoCoA (CoAP Simple Congestion Control/Advanced) [9] . In [8], a simulation had been implemented over Contiki OS in order to evaluate the performance of different alternative CoAP congestion control mechanisms.

Message reliability is done on the application layer by marking the message as confirmable in the CoAP header. This has the advantage of not overloading the network with the acknowledgement of non-critical messages and hence preserve network bandwidth.

B. MQTT (Message Queuing Transport Protocol)

Unlike CoAP, MQTT is based on the subscriber/publisher architecture. The main objectives of MQTT is the connection with remote locations where constrained devices is required and/or bandwidth cost is high. Figure 3 shows the MQTT model and protocol stack:

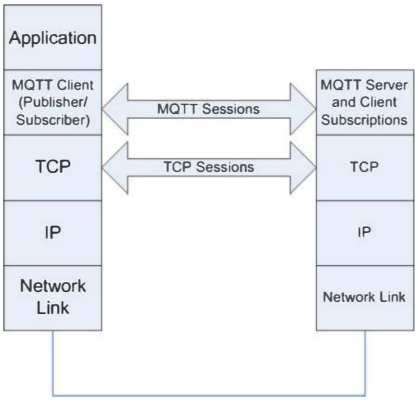

Fig. 3. MQTT Model and Protocol Stack and session details

Figure 4 shows the packet structure of MQTT protocol. The protocol overhead is 2 mandatory bytes with optional variable length header (1-4) bytes.

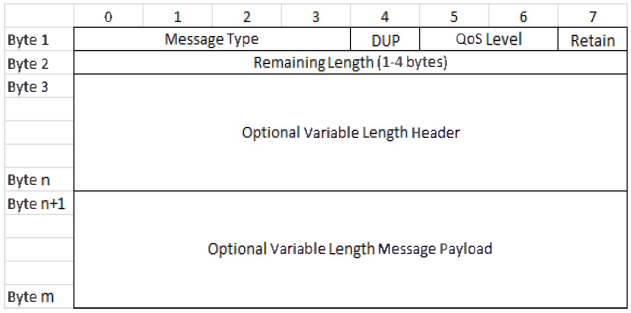

Fig. 4. MQTT Packet

As MQTT is based on TCP which acknowledges all the messages on the transport layer. This increases the load on the network. There are three levels of QoS in MQTT messages 1) At most one: The minimal level is zero and it guarantees a best effort delivery. A message won't be acknowledged by the receiver or stored and redelivered by the sender. This is often called "fire and forget" and provides the same guarantee as the underlying TCP protocol. 2) At least once: it is guaranteed that a message will be delivered at least once to the receiver. But the message can also be delivered more than once. 3) Exactly once: it guarantees that each message is received only once by the counterpart. It is the safest and also the slowest quality of service level. The guarantee is provided by two flows there and back between sender and receiver. [10]

C. XMPP (Extensible Messaging and Presence Protocol)

XMPP is a near real-time communication protocol that relies on Extensible Mark-up language (XML) and enables the exchange of structured data between network entities. XMPP uses decentralized client-server model, where each use connects to the server that controls its own domain. Thus, allowing the creation of interoperable and federated architecture based on multiple authorities [11].

A XMPP session is established after creating a TCP connection, XMPP exchange input/output XML streams for opening the channel that is going to be used during all communication process. XML stream is a container for exchanging between entities. The stream is started by sending header tag stream [11]. XMPP is a general protocol for near real-time messaging, and request/response services.

Network overhead is one of the main problems of XMPP because of sending the data between entities in text format in the XML stream. XML depends on TCP for message reliability which increases the network overhead.

Figure 5 shows the architecture of the XMPP protocol.

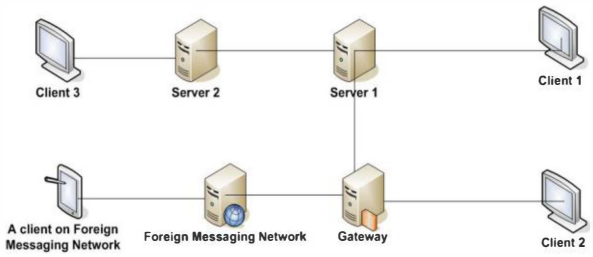

Fig. 5. XMPP Architecture

III. M2M Device Management Protocols

A. OMA DM:

The OMA DM protocol is designed by OMA and mainly targeting the device management of mobile devices such as mobile phones, and tablets. The features of the OMA DM protocol is to read, write configuration or monitoring nodes. Also, it includes the triggering remote commands, firmware update and software component management object.

Also the OMA DM is originally designed for mobile phones and similar devices, it can be used with M2M devices. OMA DM is the de facto standard for mobile device management. The OMA DM protocol supports the functions of 1) provisioning 2) configuration maintenance and management 3) software management 4) fault detection, query and reporting 5) non-application software download 6) configuration of user preference. [12]

OMA DM specification includes management information for mobile devices in the form of DM tree [13,14,15] and management protocol for remotely managing mobile devices. Figure 6 shows the OMA DM standard management architecture and protocol transaction details

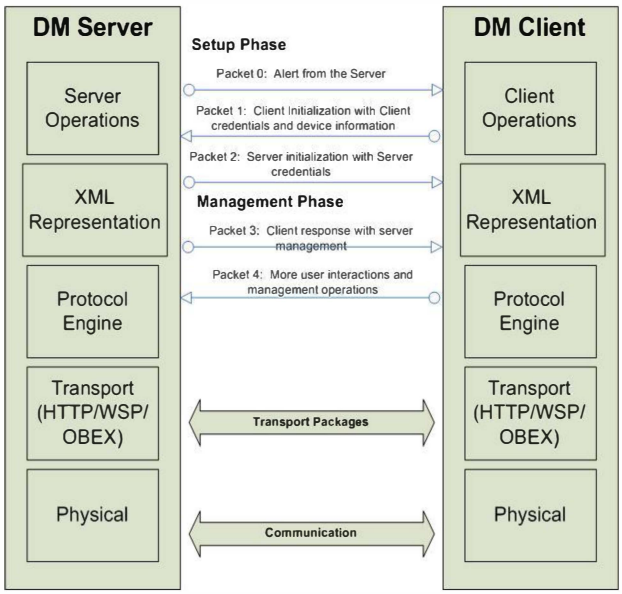

Fig. 6. OMA DM standard architecture and protocol details

B. LwM2M (lightweight M2M)

LwM2M is the successor of OMA DM protocol for M2M devices and aimed to be more light than OMA DM. LwM2M specifications defines how the LwM2M client communicate with LwM2M server in the application layer. The LwM2M client is located on the M2M device. The main target of LwM2M is the device management and service enablement for the M2M devices.

The LwM2M client is located on the M2M device and communicates with the LwM2M server which resides on the M2M Service Provider or the Network Service Provider and serves as endpoint of the LwM2M protocol. LwM2M supports multiple servers.

LwM2M protocol stack utilizes the CoAP as the underlying transfer protocol over UDP or SMS bearers. CoAP defines the message header, request/response codes, message options, and retransmission mechanisms. The overhead of LwM2M is the overhead of CoAP plus the application objects such as JSON (Javascript Object Notation) which is 10s of bytes. This is compared to the overhead of OMA DM protocol which consists of HTTP plus XML messages which results in 100s of bytes [16]

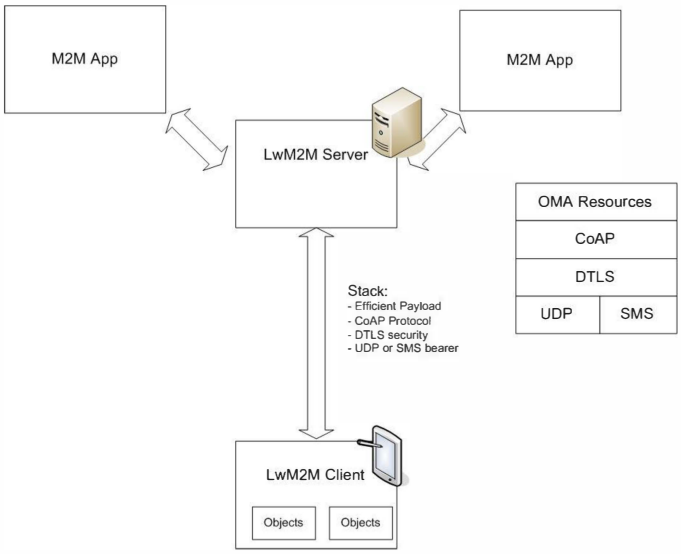

Fig. 7. LwM2M Architecture and Protocol Stack

The LwM2M protocol provides different features such as firmware upgrade, device monitoring and configuration, and server provisioning. LwM2M can be used to wake-up the devices via SMS or any get/post/put/delete commands. The device can reply back via SMS or UDP. Figure 7 shows the general architecture of the LwM2M protocol.

As the LwM2M is based on top of CoAP and UDP, it does not provide message reliability like OMA DM. However, it is more lighter than OMA DM for the decreased headers and binary encoding.

C. Technical Report – 069

TR-069 describes the CPE WAN Management Protocol (CWMP), intended for communication between a CPE

(Customer Premises Equipment) and ACS (Auto Configuration Server). The CWMP defines a mechanism that encompasses secure auto-configuration of a CPE and also incorporates other CPE management functions into a common framework. The CWMP is using SOAP over HTTP as transport of the messages [17]. Figure 8 shows the architecture and protocol stack of CWMP.

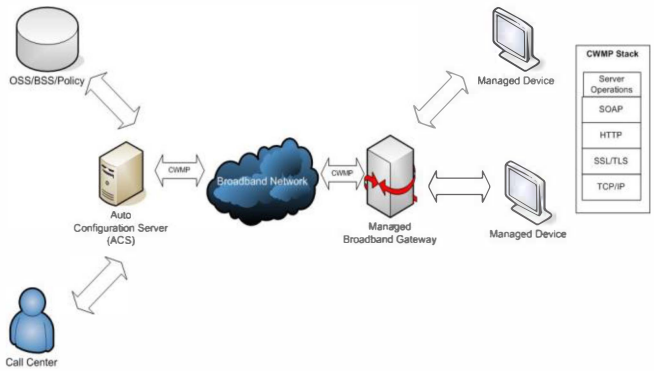

Fig. 8. CWMP protocol architecture and stack.

Although CWMP is based on TCP in the transport layer, the CWMP implements its own reliability. The CPE must receive a confirmation from the ACS in order to consider the event is successfully delivered. ACS has to be prepared to receive the same event more than one time because the ACS might sent a response that is not delivered to the CPE [17].

The CWMP protocol provides different core functionalities including: 1) Auto Configuration and Dynamic service provisioning: The CWMP allows an ACS to provision a CPE or collection of CPE. The provisioning mechanism allows the CPE provisioning at the time of the initial connection to the broadband access network, and the ability to re-provision or re-configure at any subsequent time. 2) Software/Firmware image management: The CWMP provide tools to manage downloading of software/firmware image files. The protocol provides mechanisms for version identification, file download initiation, and notification of the ACS of the success or failure of a file download. 3) Software Module Management: The CWMP enables an ACS to manage modular software and execution environments on a CPE. The capabilities include the ability to install, update and uninstall software modules as well as notification to the ACS of success or failure of each action 4) Status and Performance Monitoring: CWMP provides support for a CPE to make available information that the ACS may use to monitor the CPE's status and performance statistics. 5) Diagnostics: The CWMP provides support for a CPE to make available information that the ACS may use to diagnose and resolve connectivity or service issues as well the ability to execute defined diagnostic tests [17].

As a result of using SOAP over HTTP which implies sending the data in text format and specification of CWMP own reliability and confirmation, the CWMP is adding large overhead to the network when compared to LwM2M over CoAP.

IV. Security in M2M Protocols

Security is a very important topic for M2M. Internet of Things (IoT) generally and M2M cannot be very popular without paying attention to security. As M2M is serving many industries, the information sent by M2M can be very sensitive to the M2M customers.

A. CoAP & LwM2M Security

CoAP is providing security based on DTLS (Datagram Transport Layer Security) which runs over UDP transport layer. DTLS is a standardized security protocol designed to provide end-to-end secure communication among two peers in the presence of unreliable datagram protocols such as UDP. [18] DTLS is designed to be as much similar as possible to the widely adopted TLS protocol. [19], and provide equivalent security services i.e. it allows client and server applications to communicate with one another preventing eavesdropping, tampering, and message forgery. In order to establish a DTLS session, two peers perform a preliminary message exchange known as handshake, so agreeing on a cipher and establishing a common security material [18].

The DICE (DTLS in Constrained Environment) work group is initiated by IETF to define a DTLS profiles that is suitable for IoT applications and implementable on many constrained devices. Also, the objective of the group is to define how DTLS record layer can be used to transmit multicast messages securely [20].

B. MQTT Security

MQTT security is based on the TLS/SSL (Transport Layer Security/Secure Socket Layer) to provide transport encryption. It provides a security against eavesdropping.

On the application layer, MQTT application provides client identifier and username/password credentials which can be used to authenticate devices on the application level. These properties are provided by the protocol itself. The disadvantage of MQTT security is the use of TLS/SSL which is not optimized for constrained devices (i.e. devices with limited processing power, memory, etc.) [21]

C. XMPP Security

XMPP includes a method for securing the stream from tampering and eavesdropping. This channel encryption method make use of the TLS. The administrator of any M2M domain may use TLS for client-to-server communications, server-to-server communications, or both. Clients should use TLS to secure the streams prior to attempting the completion of the security negotiation [22]. Similar to MQTT security model, The disadvantage of using the TLS is that it is not optimized for the constrained devices.

D. OMA DM Security

OMA DM security is providing different security services including 1) Credentials: using username/password 2) Authentication: Both OMA DM client and server must be authenticated to each other on different layers (i.e. transport layer authentication, or application layer in case transport layer

authentication is not supported by the client). The authentication is based on MD5 authentication. The use of transport layer that supports encryption is not mandatory but recommended [23]. When using OMA DM over HTTP, TLS/SSL must be supported.

E. CWMP Security

CWMP security is designed to prevent tampering with the transactions that take place between CPE and ACS. The mechanisms incorporated in this protocols are 1) TLS to provide transaction confidentiality, and data integrity. 2) HTTP layer provides an alternative means of CPE and ACS authentication based on shared secrets. The protocol does not specify how the shared secrets are learned by CPE and ACS [17].

V. CONCLUSION

In this paper, we have undertaken a comprehensive survey of the different M2M protocols used for M2M communication and Device Management. CoAP is providing very small network overhead beside laying on DTLS for security which is designed especially for constrained devices and provide the same security services as TLS. LwM2M works over CoAP and both provides a coupled protocol for both communication and Device management. MQTT is based subscriber/publisher architecture and provide tiny network overhead. Using TCP means acknowledgement of each packet which increase the network overhead especially that lots of M2M machines are using 2G connects due to power constraints. The security of MQTT is based on TLS/SSL which is not optimized for constrained devices. XMPP uses XML for data communication. This put overhead on the network for not supporting binary formats. The use of TLS increases the overhead. OMA DM protocol is originally designed for smartphones which are not so limited in resources like M2M devices. The OMA published another standard for the LwM2M which is more lighter than OMA DM. The security of OMA DM is mandatory when using HTTP by using TLS/SSL. The CWMP protocol provide different functionalities for device management and configuration. The CWMP using SOAP over HTTP which puts high overhead over the network. The security of CWMP is based on TLS and shared secrets.

REFERENCES

[1] M. Zubair, Lusheg Ji, Alex X. Liu, Jeffrey Pang, and Jia Wang, "Large-Scale Measurement and Characterization of Cellular machine-to-machine Traffic", "IEEE/ACM Transaction on Networking (TON), NJ, USA, Vol. 21, Issue 6, pp 1960-1973, December 2013

[2] "LMU-2600 GPRS Fleet Tracking Unit", 2012 [online], available: http://www.calamp.com/pdf/LMU-2600.pdf

[3] A. Betzler, C. Gomez, I. Demirkol, J. Paradells, "Congestion Control in Reliable CoAP Communication", "MSWiM '13 Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems", NY, USA, pp 365-372, November 2013

[4] E. Vergara, M. Prihodko, S. Tehrani, "Mobile Location Sharing: an energy consumption study", "e-Energy '13 Proceedings of the fourth international conference on Future energy systems", NY, USA, pp. 289-290, January 2013

[5] A. Elmangoush, A. Al-hezmi, T. Magedanz, "Towards Standard API for Cloud-Based Telco Service Platforms", "MSWiM '13 Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems", NY, USA, pp 365-372, November 2013.

[6] CoAP", 2014 [online], available: http://coap.technology/

[7] Z. Shelby, K. Hartke, C. Bormann, "The Constraint Application Protocol (CoAP)", RFC 7252 (Proposed Standard), June 2014

[8] A. Jara, P. Lopez, D. Fernandez, J. Castello, M. Zamora, A. Sharmeta, "Mobile Discovery: discovering and interacting with the world through the internet of things", "Personal and Ubiquitous Computing", Springer-Verlag London, UK, pp 323-338, February 2014

[9] A. Betzler, C. Gomez, I. Demirkol, "Evaluation of Advanced Congestion Control Mechanisms for Unreliable CoAP communications", PE-WASUN'15 proceeding of the 12th ACM symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks, NY, USA, pp 63-70, November 2015.

[10] Enterprise MQTT Broker, 2015 [online], available http://www.hivemq.com/

[11] H. Flores, S. Srirama, "Mobile Cloud Messaging Supported by XMPP Primitives", "MCS '13 Proceeding of the fourth ACM workshop on Mobile cloud computing and services", NY, USA, pp 17-24, June-2013

[12] J. Kang, H. Ju, M. Choi, J. Hong, J. Kim, "OMA DM-based remote software fault management for mobile devices", "International Journal of Network Management", NY, USA, Vol. 19, Issue 6, pp 491-511, November 2009.

[13] OMA Device Management Standardized Objects, July 2008, Published Online,Available: http://technical.openmobilealliance.org/Technical/Release_Program/docs/DM/V1_2_1-20080617-A/OMA-TS-DM_StdObj-V1_2_1-20080617-A.pdf

[14] OMA DM Device Description Framework, January 2005, Published Online,Available: http:// member.openmobilealliance.org/ftp/public_documents/dm/Permanent_documents/OMA-TS-DM-DDF-V1_ 2_0–20050131-D.zip

[15] OMA DM Device Management Tree and Description, January 2007, Published Online, Available: http:// member.openmobilealliance.org/ftp/public_documents/dm/Permanent_documents/OMA-TS-DM_TND-V1_ 2–20070115-C.zip

[16] G. Klas, F. Rodermund, Z, Shebly, S. Akhouri, J. Holler, "Lightweight M2M: Enabling Device Management and Applications for the Internet of Things", White Paper, Feb 2014

[17] S. Banks, A. Colmegna, T. Spets, "CPE WAN Management Protocol", Issue 1, Amendment 4, July 2011

[18] M. Tiloca, "Efficient Protection of Response Messages in DTLS-Based Secure Multicast", "SIN '14 Proceedings of the 7th International Conference on Security of Information and Networks", Glagsow, Scotland, pp 466, September 2014

[19] T. Dierks and E.Rescorla, RFC 5246, The Transport Layer Security Protocol Version 1.2, Internet Engineering Task Force, August 2008.

[20] D. Gellert, Z. Shelby, "DTLS in Constrained Environments (Dice)", Available: https://datatracker.ietf.org/wg/dice/charter/

[21] Introducting the MQTT Security Fundamentals, 2015 [Online], Available: http://www.hivemq.com/blog/introducing-the-mqtt-security-fundamentals

[22] P. Saint Andre, RFC 3920, "Extensible Messaging and Presence Protocol (XMPP): Core", Internet Engineering Task Force March 2011.

[23] OMA DM Device Management Security, June 2008, Published Online,Available: http://technical.openmobilealliance.org/Technical/Release_Program/docs/DM/V1_2_1-20080617-A/OMA-TS-DM_Security-V1_2_1-20080617-A.pdf